CSCE 230 – Lab 2: Basic Assembly Program

Due: 11:59PM, Sept 5 (Tuesday)

Objectives

- ❖ Learn the basic syntax and structure in Nios II assembly programming
- ❖ Learn how to use loops

Useful References on Canvas

- ❖ Lecture notes for Chapter 2 and Appendix B
- ❖ Altera Nios II Processor Document
- ❖ Altera Nios II Instruction Document

Lab: Basic Assembly Program

First, please create a new project

- Create a new project directory. If you are using the lab computers, it is more convenient to create a new project directory in C:\temp
- In your project directory, create an empty text file using any text editor, and name it lab2.s or any other names, but the file extension should be .s
- Open the Altera Monitor program
    - Menu “File” → “New Project”
    - In the “New Project Wizard”,
        - Select your project directory and specify a project name, and then click “Next”.
        - Select system “DE10 Light Computer”, and then click “Next”
        - Select program type “Assembly Program”, and then click “Next”
        - Add your lab2.s file into the project, and then click “Next”
        - Use the default system parameters, and then click “Next”
        - Use the default/basic memory settings (This will put the .text section at 0 and .data would immediately follow in memory), and then click “Finish”

Then, open your lab2.s file to write a program to find the minimum positive integer n such that function r3(n) is greater than 10. Function r3(n) is defined as follows

$$r3(n) = \sum_{r4=1}^{n} \left(\sum_{r5=1}^{r4} r5 \right)$$

When the program finishes, the value of register r4 should be the minimum n.
To help you understand this function, the following code fragment shows how to find the minimum n such that function r3(n)>10 in language C/Java. You will submit on Canvas when complete.

```
r2 = 10;
r3 = 0;
r4 = 1;
do
{
    for(r5=1; r5<=r4; r5=r5+1)
    {
        r3 = r3+r5;
    }
    r4 = r4+1;
} while(r3<=r2);
r4 = r4-1;
```

Please answer the following questions on Canvas.

- Questions 1-10: Please fill in the ten blanks (using lowercase letters or numbers) in the following assembly program, which is very similar to the previous C/Java program.

```
 .text
    .global  _start
_start:
    movi    r2, 10
    movi    r3, 0
    __1_    r4, 1
dowhile:
    movi    r5, 1
forloop:
    __2_    r3, r3, r5
    addi    r5, r5, 3_
    ble     4_, 5_, forloop
    __6_    r4, 7_, 1
    __8_    9_, r2, dowhile
    subi    r4, r4, 10
end:br      end
    .data
    .end
```

Quick correctness check:

- Please set a break at label "end", and run the program. When the program stops at label "end", the value of register pc should be 0x00000028, and the value of register r3 should be 0x00000014 in hexadecimal or 20 in decimal.

Please continue to answer the following questions on Canvas

- Questions 11-12: When the program stops at label "end", what are the values of register r4 and r5 (in hexadecimal, with 0x followed by exactly 8 hexadecimal symbols)?
- Question 13: What is the word (in hexadecimal, with 0x followed by exactly 8 hexadecimal symbols) at memory address 0?
    - This information can be found just in the "Disassembly window", where there are three columns. The first column shows the memory addresses, the second column shows the word (i.e., the binary machine instruction) at each address, and the third column shows the corresponding assembly and machine instructions.
- Question 14: What is the byte (in hexadecimal, with 0x followed by exactly 2 hexadecimal symbols) at memory address 0?
    - This information can be found from your answer to Question 13.
    - Recall that a Nios II processor uses byte addressing (i.e., successive addresses are assigned to successive bytes) and little-endian (the first byte in a word is the little end, that is, the least significant end).
- Question 15: What is the byte (in hexadecimal, with 0x followed by exactly 2 hexadecimal symbols) at memory address 6?
    - This information can be found just in the "Disassembly window"
    - Recall that a Nios II processor uses byte addressing (i.e., successive addresses are assigned to successive bytes) and little-endian (the first byte in a word is the little end, that is, the least significant end).
- Question 16: Submit a copy of your lab2.s assembly program to find the minimum positive integer n such that function r3(n) is greater than 10.?
    - This is the lab2.s code you created for the first part of this lab.